

CRICKET WORLD CUP

(Team Name: Collaborative Analysts)

Introduction

The mini-world for our project is going to be the Cricket World-Cup.

Cricket is the most popular sport in our country and we know that the 2021 Cricket T-20 World cup is coming up in a few weeks. A lot of us would be watching the tournament and thus we would like to be kept updated about the tournament.

The tournament would start from 11th sept ,2021 and the finals would be played on 6th Oct,2021.

In total, 10 teams would be participating in the tournament and it will be played by a round-robin followed by knockout format. After the league stages where teams would play against every other time 2 times , the top four teams would be selected for the semi-finals which would be played in a knockout manner.

Purpose

This database would represent, store and update all the data about the tournament.

We would be able to design the tournament by inserting what teams are playing, which players would be playing for each team, their statistics, contract, sponsors, venue for the matches , etc.

This datatable would be a real time representation of the tournament so we would be able to design it the way we want by choosing which match would be played when, where, between which two teams, etc and we would also update the data alongside .

Users

Different users can use the database in the following manner:

1) Players: Players can use the database to check which matches are going to be played in which venues, their statistics over the course of the tournament and also the points table to help them keep track of their performance.

2) Organizers: Organizers would be able to know the attendance for each match and thus they would be able to calculate the revenue generated and this would help them to prepare and make changes if required.

3) Sponsors: They can get info about which brands they are competing against . This would help to decide in which venues they should invest their money in order to generate profit.

4) Viewers: They can see all the statistics about the tournament like how well is their favorite team/player performing, what interesting matches are gonna be played in the coming days, points table ,etc.

Applications

The different applications of the database are given as follows :

- 1) Different companies can access the database and create apps that would help the audience be up to date with the current affairs of the tournament.
- 2) Sponsors can access the database to get to know the teams and players and offer brand deals.
- 3) Travel companies can access the database and get to know the match schedules to arrange transportation respectively.
- 4) Franchise Finance heads can access the database to set Ticket prices for different matches depending on the venue and percentage turnover.

Database Requirements

Entities

1. Matches

- a. Match_ID
int(101-999),Primary key constraint
- b. Team_1_ID
int(1-10),NOT NULL constraint
- c. Team_2_ID
int(1-10),NOT NULL constraint
- d. Match Schedule
(Composite -> Contains date as well as time)
First value: DD/MM/YYYY ,NOT NULL constraint,Check constraint (1 <= DD <= 31 ; 1 <= MM <= 12 ; 1976 <= YYYY <= 2003)

Second value: Starting Time of the match-> HH:MM where HH is the hour time (24-hour format) and MM is the minute at which the match starts,NOT NULL constraint,Check constraint(HH : 00<=HH<24 , MM: 00<=MM<60)

- e. Venue_ID
int(100-999), NOT NULL constraint
- f. Win_Type
(Composite->Contains Winnner_Team_ID and Won_by)
First value: int(1-10) , NOT NULL constraint
Second value: varchar(15) , NOT NULL constraint
Second value tells by how many runs or wickets remaining the winning team won.
So input would be like "20 runs" or "5 wickets".
- g. Man_of_the_Match-Player_ID
int(100,999), NOT NULL constraint
- h. Match_Attendance
int(0-100000), NOT NULL constraint
- i. Percentage_Turnover(DERIVED ATTRIBUTE)
(Match_Attendance/Venue_Capacity)
float(2,2 (digits before and after decimal point) , NOT NULL constraint, CHECK constraint(0-100)
- j. Sponsors
Multivalued Attribute since a match can have multiple sponsors.
varchar(100), NOT NULL constraint

2. Players

- a. Player_ID
int(5 digit), Primary Key constraint
- b. Player_Name
String, NOT NULL constraint
- c. Team_ID
int(1-10), NOT NULL constraint
- d. Role
varchar, NOT NULL constraint, has to be one of Batsman, Bowler or All_Rounder
- e. DOB
First value: DD/MM/YY , NOT NULL constraint, Check constraint (1 <= DD <= 31 ; 1 <= MM <= 12 ; 1976 <= YYYY <= 2003)
- f. Player_Age (Derived from DOB)
int(18 - 45), NOT NULL constraint
- g. Phone_No
Multivalued Attribute (a player can have multiple phone numbers), varchar

3. Venue

- a. Venue_ID
int(101 - 999), Primary Key Constraint
- b. Venue Name
varchar(50), NOT NULL Constraint
- c. Stadium Name
varchar(50), NOT NULL Constraint

- d. Stadium Capacity
int(1-50000), NOT NULL Constraint
- 4. Team
 - CANDIDATE KEY: Ranking
 - SUPER KEY: {Team_ID,Team_Captain}
 - ALTERNATE KEY: Ranking
 - a. Team_ID
int(1-10), Primary Key Constraint,
 - b. Team_Captain
char(50), NOT NULL Constraint
 - c. Team_Coach
char(50), NOT NULL Constraint
 - d. Country_Name
char(50), NOT NULL Constraint
 - e. Ranking
int(1-20), NOT NULL Constraint, Unique Constraint
- 5. Batting Statistics
 - a. Player_ID
int(5 digit), foreign key constraint
 - b. Highest_Score
int(3 digit), NOT NULL constraint
 - c. Batting_Average
float(3,2), NOT NULL constraint
 - d. Batting_Strike_Rate
float(3,2), NOT NULL constraint
 - e. Number_of_4s
int(1-1000), NOT NULL constraint
 - f. Number_of_6s
int(1-1000), NOT NULL constraint
 - g. Batting_style
Varchar, NOT NULL constraint
- 6. Bowling Statistics
 - a. Player_ID
int(5 digit), foreign key
 - b. Bowling_Average
Float, NOT NULL constraint
 - c. Max_Wickets_Taken
int(0-10), Check constraint(<=10).
 - d. 5_Wickets
int(0-100), NOT NULL constraint.

- e. Bowling_style
varchar(50), NOT NULL constraint
- 7. Points Table
 - a. Team_ID
int(1-10), primary key constraint
 - b. Team_Name
varchar(50), NOT NULL
 - c. Points_Scored
int(0-100), NOT NULL
 - d. Run_Rate
float(3,2), NOT NULL
 - e. Ranking
int(1-10), Primary Key Attribute,(candidate key), Check constraint (<=10)

Weak Entity:

Batting Statistics & Bowling Statistics are weak entities because these can be the same but unique for a player i.e. a single player cannot have two batting/bowling statistics. These can be uniquely identified by Player_ID hence Player is the identifying relationship type for these two entity types.

Subclass:

Batsman, Bowler & All_Rounder are subclasses of Player.

- 1. Batsman
 - a. Total_Runs_Scored
int(0-20000), NOT NULL constraint
 - b. No_of_100s
int(1-100), NOT NULL constraint
 - c. No_of_50s
int(1-100), NOT NULL constraint
 - d. Highest_runs
int(1-1000), NOT NULL constraint
 - e. Average
float(3,2), NOT NULL constraint
- 2. Bowler
 - a. Total_Wickets_Taken
int(0-1000), NOT NULL constraint
 - b. No_of_3_wickets
int(0-1000), NOT NULL constraint
 - c. No_of_5_wickets
int(0-1000), NOT NULL constraint
 - d. Highest_wicket

- int(0-100), NOT NULL constraint*
 - e. Economy
 - float(3,2), NOT NULL constraint*
- 3. All_Rounder
 - a. Total_Runs_Scored
 - int(0-20000), NOT NULL constraint.*
 - b. Total_Wickets_Taken
 - int(0-1000), NOT NULL constraint*
 - c. Highest_runs
 - int(0-500), NOT NULL constraint*
 - d. Highest_wickets
 - int(0-10), Check constraint(<=10), NOT NULL constraint*
 - e. Batting_Average
 - float(3,2) , NOT NULL constraint*
 - f. Bowling_Average
 - float(3,2) , NOT NULL constraint*
 - g. Economy
 - float(2,2), NOT NULL constraint*

Relationships:

1. Contract
 - a. Degree : 4
 - b. The relationship is between Matches, Players, Venue, Team.
 - c. Cardinality Ratio : 1:N:1:2
 - d. Participation Constraints:

Matches(1,1),Players(N,N),Venue(1,1),Team(2,2) All have total participation.
2. Belongs To

Player **Belongs To** Team.

 - a. Degree: 2
 - b. Relationship between Player & Team
 - c. Cardinality Ratio: N:1
 - d. Player: (1,N),Team:(1,1), Team & Player both have total participation
3. Plays

Player **Plays** Matches

 - a. Degree: 2
 - b. Relationship between Player & Matches
 - c. Cardinality Ratio: N:1

- d. Player: (1,N), Matches:(1,1), Player has partial participation, Matches have total participation

4. Is played at

Match **is played at** Venue

- a. Degree: 2
- b. Relationship between Match & Venue
- c. Cardinality Ratio: N:1
- d. Match: (1,N), Venue: (1,1)

5. Match Schedule

Match Schedule is decided by Matches, Venue & Teams playing in a match.

- a. Degree: 3
- b. Relationship between Matches, Venue, teams
- c. Cardinality Ratio: 1:1:N
- d. Matches: (1,1), Venue: (1,1), teams (2,N).

n>3 Relationships:

Contract :

Relationship between the following entities : Matches, Players, Venue, Team. The attributes that the Relationship will contain are Match_ID, Venue Name, Player_ID, Team_ID, Stadium Name.

Before every match , each 'player' of the 'teams' which are playing will sign a contract for the 'Match' is going to be played in a specific 'venue' and for signing the contract,the player will also receive some payment for signing the contract.

Hence,there will also be an added attribute of Match_Fees for every player signing the contract for a given match.

Functional Requirements

Modifications

1. Insert: Inserting Player Details,Match Details, Venue details etc.
2. Delete: Deleting Player, Venue etc.
3. Update: Updating Player Details, Match Details etc.

Retrievals

1. Selection: Get all player details from a particular team.
2. Projection: Get results of all matches of a particular team.
3. Aggregate: Average runs scored in matches at a particular venue
4. Search: Get all player names starting with "ZAH"
5. Analysis: i) Highest score by a player for a particular team.
ii) Team which scored Highest no of boundaries.